



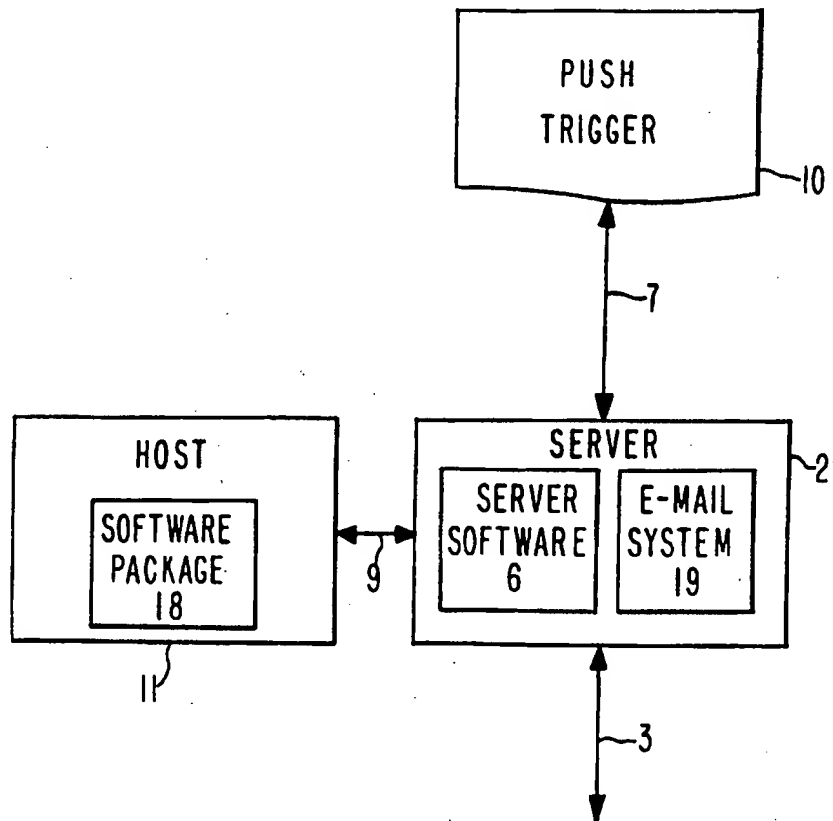
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : G06F 9/445	A1	(11) International Publication Number: WO 98/53397 (43) International Publication Date: 26 November 1998 (26.11.98)
(21) International Application Number: PCT/US98/08784 (22) International Filing Date: 29 April 1998 (29.04.98) (30) Priority Data: 08/861,177 21 May 1997 (21.05.97) US (71) Applicant: SYMANTEC CORPORATION [US/US]; 10201 Torre Avenue, Cupertino, CA 95014 (US). (72) Inventor: SADOWSKY, Richard; 4873 Tampico Way, San Jose, CA 95118 (US). (74) Agents: RADLO, Edward, J. et al.; Fenwick & West LLP, Two Palo Alto Square, Palo Alto, CA 94306 (US).		(81) Designated States: CA, JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>With international search report.</i>

(54) Title: PUSH DEPLOYMENT OF SOFTWARE PACKAGES USING NOTIFICATION TRANSPORTS

(57) Abstract

A system and method for deploying one or more software packages (18) over a communications interface (1). A push trigger (10) is sent to a server computer (2). In response to the trigger (10), the server computer (2) generates a notification package (12, 13, 14). The notification package (12, 13, 14) is sent over the communications interface (1) to at least one client computer (2). Each client computer (2) contains at least one notification transport (15, 16, 17) which recognizes a corresponding notification package (12, 13, 14). The notification transport (15, 16, 17) then instructs the server computer (2) to automatically push the software package (18) to the client computer (4) over the communications interface (1). E-mail transport (15), UNC transport (16), and opportunistic transport (17) are described.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav	TM	Turkmenistan
BF	Burkina Faso	GR	Greece		Republic of Macedonia	TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's	NZ	New Zealand		
CM	Cameroon		Republic of Korea	PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

DESCRIPTION

**PUSH DEPLOYMENT OF SOFTWARE PACKAGES
USING NOTIFICATION TRANSPORTS**

RELATED APPLICATION

5 A related patent application is U.S. patent application
Serial No. 08/664,388 filed June 14, 1996 by Richard Scott
Sadowsky entitled "Use of Polymorphic Package Files to Update
Software Components", which patent application is hereby
incorporated by reference in its entirety into the present
10 patent application.

TECHNICAL FIELD

This invention pertains to the field of using push
technology to deploy software packages from at least one server
computer to at least one client computer.

15

BACKGROUND ART

A software product known as BackWeb is a product that
manages software updates using push installation. However,
unlike the present invention, BackWeb uses an HTTP connection,
which requires access to the World Wide Web component of the
20 Internet. The present invention is not so limited. For
example, the present invention can work on top of conventional
e-mail systems in environments where access to the World Wide
Web is not available. E-mail is currently used by over 70
million users worldwide; and provides for a fast, convenient,
25 automatic, and inexpensive way to deploy software updates.
Furthermore, the present invention employs true push technology
unlike BackWeb, which pulls on a given "channel", i.e., BackWeb
looks at a given URL (Universal Resource Locator). In the e-
mail embodiment of the present invention, messages are queued
30 and processed efficiently by the e-mail post office, and the
user need not maintain constant access to the Internet. Another

advantage of the present invention over BackWeb is that the BackWeb client software is over 4 megabytes in size, whereas the present invention's agent software is under one megabyte.

Finally, the present invention, unlike BackWeb, does not subject the user to viewing vendors' advertisements.

DISCLOSURE OF INVENTION

The present invention is a system and method for deploying a software package (18) over a communications interface (1). The method comprises the steps of sending a trigger (10) to a server computer (2). In response to the trigger (10), the server computer (2) generates a notification package (12, 13, 14). The notification package (12, 13, 14) is sent over the communications interface (1) to at least one client computer (4). The notification package (12, 13, 14) is recognized by a notification transport (15, 16, 17) located within each client computer (4). The notification transport (15, 16, 17) then instructs the server computer (2) to automatically push the software package to the client computer (4) over the communications interface (1).

BRIEF DESCRIPTION OF THE DRAWINGS

These and other more detailed and specific objects and features of the present invention are more fully disclosed in the following specification, reference being had to the accompanying drawings, in which:

Figure 1 is a block diagram of a computer system architecture that may advantageously use the present invention.

Figure 2 is a block diagram of components on server computer 2 side of communications interface 1 of the present invention.

Figure 3 is a flow chart illustrating method steps that are performed within server computer 2 in a preferred embodiment of the present invention.

Figure 4 is a block diagram illustrating method steps that
5 are performed within client computer 4 in a preferred embodiment of the present invention.

Figure 5 is a block diagram illustrating a more general embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Figure 1 illustrates an operating environment in which the present invention may advantageously be employed. One or more server computers 2 are coupled to a communications interface 1 via links 3. Communications interface 1 may be any communications interface, such a direct modem connection, the Internet, an intranet, an extranet, or any combination thereof (mixturenet).

Within each server computer 2 is a server software module 6 that has been designed to perform method steps of the present invention. Also coupled to communications interface 1 via links 5 are one or more client computers 4. Within each client computer 4 is an agent software module 8 that has been designed to perform method steps of the present invention.

Links 3 and 5 can be electrically conductive wires, fiber optic cables, wireless links, or any combination thereof. Furthermore, links 3 and 5 can be local or remote links.

Computers 2 and 4 can be any type of computer, and are typically digital computers. In any given system, computers 2 may be similar or identical to each other or may be different. Similarly, computers 4 may be similar or identical to each other or to server computers 2, or may differ therefrom. Server modules 6 and/or agent modules 8 may be implemented in firmware, hardware, and/or software, but for purposes of simplicity will normally be referred to herein as software modules.

Figures 2, 3, and 4 illustrate a preferred embodiment of the present invention in which the notification transport is an e-mail (electronic mail) transport 15.

Figure 5 illustrates a more general embodiment of the present invention in which a UNC (Universal Naming Convention) transport 16 and an opportunistic transport 17 are also shown.

Figure 2 illustrates components that are typically present on the server computer 2 side of communications interface 1. Push trigger 10 is a message that is activated upon a certain event, or combination of events, e.g., the end of the month, the presence of new anti-virus information posted within host computer 11, etc. Trigger 10 can comprise software that has been preprogrammed to trigger upon said event or events. Alternatively, trigger 10 can be generated by a human operator activating some device, such as a computer keyboard.

- 10 - Trigger 10 may include a set of addresses corresponding to client computers 4 that are intended to receive the corresponding one or more software packages 18.

Trigger 10 is sent over link 7 to server computer 2. Link 7 may be a remote or local link, and may be wired, wireless, or comprise fiber optic cables. Host computer 11 may be part of server computer 2, may be identical with server computer 2, may encompass server computer 2, or may be a standalone computer. Host computer 11 contains one or more software packages 18 that will be deployed to one or more client computers 4.

- 20 Each software package 18 may contain a computer program, a software patch, new data, or any combination of these items. A given software package 18 may comprise polymorphic software. When host computer 11 is separate from server computer 2, host computer 11 and server computer 2 communicate with each other via link 9, which may be a remote or local link, and may be wired, wireless, or comprise fiber optic cables. In the embodiment illustrated in Figures 2, 3, and 4, server computer 2 contains an e-mail system 19 (which may be an existing system) typically including storage, protocols, address tables, and communications links. For example, e-mail system 19 may be a MAPI (Messaging Application Programming Interface) e-mail system
- 25
- 30

used in conjunction with a Windows 95 or a Windows NT operating system.

Figure 3 illustrates method steps performed by server software 6. At step 60, software 6 is activated and waits for trigger 10. In an alternative embodiment, software 6 could be dormant and could be activated by trigger 10. At step 61, software 6 receives trigger 10 over link 7. At step 62, software 6 inquires as to whether trigger 10 is a push trigger. Step 62 could be implemented, for example, by having a pre-established field within trigger 10 of predetermined size, where a predetermined sequence of bits indicates a valid push trigger. If trigger 10 is not a valid push trigger, then processing is terminated at step 63. If, on the other hand, trigger 10 is a valid push trigger, step 64 is executed.

At step 64, optional but highly desirable authentication and validity substeps are performed. For example, the authentication substep can establish that a legitimate person and not a spoofer originated push trigger 10. This can be accomplished by having previously encrypted all or a portion of push trigger 10 and having the authentication substep decrypt the previously encrypted information. Alternatively, the authentication could be performed by cryptographic authentication, e.g., by a method of digital signatures. The digital signature ties the contents of a specific software package 18 with the contents of a specific push trigger 10. The digital signature is added to trigger 10. After arrival of trigger 10, software 6 (at step 64) reads the digital signature from trigger 10 and recomputes the digital signature based upon the contents of trigger 10 and the contents of software package 18. If the computed digital signature matches the digital signature stored in trigger 10, then trigger 10 is authentic. If not, trigger 10 is a forgery.

The digital signature can be computed using standard encryption algorithms of arbitrary strength. A message digest can first be created using a one-way hash function such as MD5 or MD52. The message digest is then signed by applying a
5 encryption routine, for example, a block cipher such as Blowfish described in Schneier, Applied Cryptography (John Wiley & Sons, Inc. 2d Ed. 1996), pp. 336-339. An advantage of using Blowfish is that it has a variable keylength. Under current U.S. law, the keylength can be arbitrarily strong for use within the
10 United States but must be limited to certain lengths for export outside of the United States. Thus, the variable keylength feature allows software 6 to be modified to provide very strong encryption where allowed by law and yet provide a weaker but exportable form of encryption to restricted countries.

15 The key used to sign the message digest is calculated by performing operations on software package 18. This binds trigger 10 to the specific software package 18. This ensures that the sending of trigger 10 will not cause access to any other server computer 2, preventing malicious programs from
20 spamming the system. Each trigger 10 must be sent or provoked with knowledge of the software package 18 that is intended.

The optional validation substep within step 64 could, for example, check the address fields within trigger 10 to ensure that these fields are of the correct length and format.

25 If the authenticity and validity substeps are not satisfactorily passed, processing is terminated at step 65. If the authenticity and validity substeps are passed, control passes to optional step 66.

The output of software 6 is one or more notification
30 packages 12, which, in this particular embodiment, are e-mail notification packages 12. Each notification package 12 contains

an address of an intended recipient 4, a notification message, and optionally the software package 18 itself.

Optional step 66 is a load balancing step. Step 66 can be useful when there are many software packages 18 to be deployed and/or many intended recipients 4. Load balancing can comprise batching like notification packages 12 together, introducing time delays into some of the notification packages 12 to avoid overload of the communications interface 1, etc.

Figure 4 illustrates method steps performed by agent software 8. In step 80, software 8 is activated and waits for receipt of the notification package 12. Alternatively, software 8 could be activated by package 12. However, in the preferred embodiment, software 8 is first activated, which is why it is referred to as agent software rather than client software.

In step 81, software 8 receives the e-mail notification package 12 over link 5.

Step 82 is then entered, wherein a determination is made as to whether the e-mail notification package 12 contains a push notification. This can be done, for example, by examining a pre-established field for a pre-established sequence of bits. If a push notification is not present, processing is terminated at step 83. If a push notification is present, control passes to step 84.

At optional but highly useful step 84, the authenticity of the e-mail notification package 12 is tested. This can be done similarly, identically, or analogously to the authenticity substep described above in conjunction with step 64. If the package 12 is found to be non-authentic, processing is terminated at step 85. If package 12 is found to be authentic, control is passed to step 86.

At step 86, software 8 determines whether one or more attachments are part of e-mail notification package 12. If the answer is yes, control is passed to step 90.

At step 90, software 8 determines whether the attachments, which in the preferred embodiment correspond to software packages 18, have been compressed, e.g., to conserve bandwidth or to speed processing time when traversing communications interface 1. If a software package 18 has been compressed, it is decompressed at step 90. Control is then passed to step 91.

At step 91, software 8 deploys software package 18 within client computer 4.

If step 86 determines that an attachment is not part of e-mail notification package 12, control is passed to step 87, in which software 8 determines whether communications interface 1 comprises the Internet. If so, control is passed to Internet utility 88. Utility 88 may be, for example, as described in the aforesaid U.S. patent application serial no. 08/664,388. The system and method described in said patent application resemble those embodied in a software feature known as LiveUpdate, which is part of the commercially successful software known as Norton AntiVirus. Utility 88 automatically sends a message to server computer 2 requesting that software package(s) 18 as identified in notification package 12 be sent to client computer 4. Server computer 2 having been programmed to comply with this request, then automatically pushes the software package(s) 18 to client computer 4.

If communications interface 1 does not comprise the Internet, push/pull utility 89 is invoked. Utility 89 performs an automatic and fast transfer and installation of software package(s) 18 via a network UNC (Universal Naming Convention). No FTP (File Transfer Protocol) or other Internet connection is required. In an embodiment of the present invention that has

been built, utility 89 is only 60K bytes in size. It works in conjunction with packages 204 and 214 of the aforesaid U.S. patent application 08/664,388. Since UNC rather than FTP is used to install the package 204, 214 directly, the process is
5 very efficient for users with fast network connections.

When utility 88 or utility 89 has completed its assigned tasks, control is passed to decompression step 90 as before.

Figure 5 illustrates a more general embodiment of the present invention in which other notification transports 16, 17
10 can be used in conjunction with or in addition to e-mail notification transport 15. Figure 5 illustrates an embodiment in which e-mail notification transport 15, UNC notification transport 16, and opportunistic transport 17 are all present within the same agent software 8. Alternatively, any particular
15 agent software 8 that is in the system can contain just one or two of the varying types of notification transports 15, 16, 17. Similarly, an e-mail notification package 12, UNC notification package 13, and opportunistic notification package 14 could all be present within the same server computer 2 or within two or
20 more different server computers 2.

UNC notification package 13 and opportunistic notification package 14 are analogous to e-mail notification package 12. Similarly, UNC notification transport 16 and opportunistic notification transport 17 are analogous to e-mail notification
25 transport 15.

In the case where a Windows operating system is used, each type of transport 15, 16, 17 typically comprises a DLL (Dynamic Link Library). DLL's 15, 16, and 17 become active when the associated agent software 8 becomes active.

30 E-mail notification has previously been described in conjunction with Figures 2, 3, and 4.

In the case of UNC notification, once UNC notification package 13 is generated by server computer 2, UNC DLL 16 examines package 13 and determines whether it contains notification of a new version of a software package 18 beyond what has already been installed within the associated client computer 4. This may be done, for example, by DLL 16 examining date and time fields associated with software currently residing within client computer 4, and comparing these date and time fields with corresponding fields within notification package 13. If DLL 16 determines that package 13 contains notification of a new software package 18, the contents of UNC notification package 13 are transmitted to DLL 16. DLL 16 then typically invokes push/pull utility 89 to command server computer 2 to automatically push the software package 18 to client computer 4, as described above in conjunction with e-mail notification.

Opportunistic notification works by means of opportunistic transport 17 constantly watching (patrolling) a pre-designated segment of communications interface 1. When a new connection is made to this pre-designated segment of communications interface 1 via a link 3, DLL 17 checks to see whether a new software package 18 is being offered. This can be done, for example, by DLL 17 examining date and time fields associated with software currently residing within client computer 4, and comparing these date and time fields with corresponding fields within opportunistic notification package 14. When a new software package 18 is detected, DLL 17 typically invokes push/pull utility 89 to command server computer 2 to automatically push the software package 18 to client computer 4, as described above in conjunction with e-mail notification.

The above description is included to illustrate the operation of the preferred embodiments and is not meant to limit the scope of the invention. The scope of the invention is to be limited only by the following claims. From the above

discussion, many variations will be apparent to one skilled in the art that would yet be encompassed by the spirit and scope of the present invention.

What is claimed is:

CLAIMS

1. A method for deploying a software package over a communications interface, said method comprising the steps of:

sending a trigger to a server computer:

5 in response to the trigger, generating within the server computer a notification package;

sending the notification package over the communications interface to at least one client computer;

10 recognizing the notification package by means of a notification transport located within each client computer; and

allowing the notification transport to instruct the server computer to automatically push the software package to the client computer over the communications interface.

15 2. The method of claim 1 wherein the server computer determines that the trigger is a push trigger.

3. The method of claim 2 wherein the server computer authenticates and validates the push trigger.

20 4. The method of claim 1 wherein the server computer respectively sends a plurality of notification packages to a plurality of client computers, and the server computer performs load balancing of the notification packages.

5. The method of claim 1 wherein at least one notification transport is an e-mail transport.

25 6. The method of claim 1 wherein at least one notification transport is a universal naming convention transport.

7. The method of claim 1 wherein at least one notification transport is an opportunistic transport.

8. The method of claim 1 wherein the allowing step comprises the substep of the client computer determining whether the notification package contains the software package.

9. The method of claim 8 wherein, when the client computer determines that the notification package contains the software package, the client computer deploys the software package within the client computer.

10. The method of claim 8 wherein, when the client computer determines that the notification package does not contain the software package, the client computer determines whether the communications interface comprises the Internet.

11. The method of claim 10 wherein, when the client computer determines that the communications interface comprises the Internet, the client computer asks the server computer for the software package by means of invoking an Internet module.

12. The method of claim 10 wherein, when the client computer determines that the communications interface does not comprise the Internet, the client computer asks the server computer for the software package by means of invoking a push/pull module.

13. A system for deploying software packages over a communications interface, said system comprising:

at least one server computer, each server computer linked to the communications interface and to a software package to be deployed;

at least one client computer, each client computer linked to the communications interface;

a push trigger coupled to each server computer and adapted to initiate a deployment of the software package; and

a notification transport located within each client computer, said notification transport adapted to communicate
5 with a notification package generated by a server computer in response to a push trigger.

14. The system of claim 13 wherein at least one notification transport is an e-mail transport.

15. The system of claim 13 wherein at least one
10 notification transport is a universal naming convention transport.

16. The system of claim 13 wherein at least one notification transport is an opportunistic transport.

17. A computer-readable medium containing a computer
15 program that performs a method for deploying a software package over a communications interface, said method comprising the steps of:

sending a trigger to a server computer:

in response to the trigger, generating within the
20 server computer a notification package;

sending the notification package over the communications interface to at least one client computer;

recognizing the notification package by means of a notification transport located within each client computer; and

25 allowing the notification transport to instruct the server computer to automatically push the software package to the client computer over the communications interface.

1/5

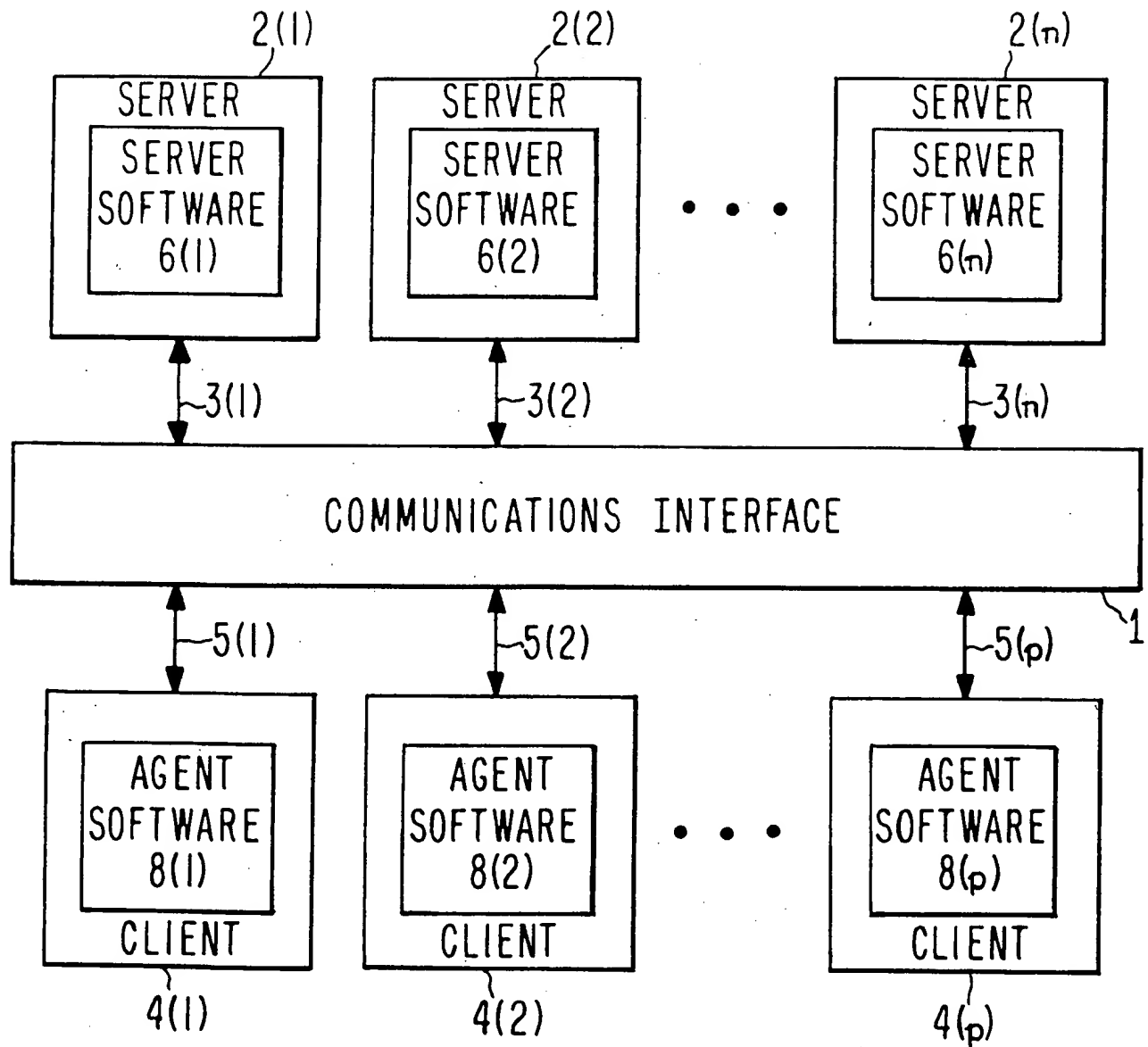


FIG. 1

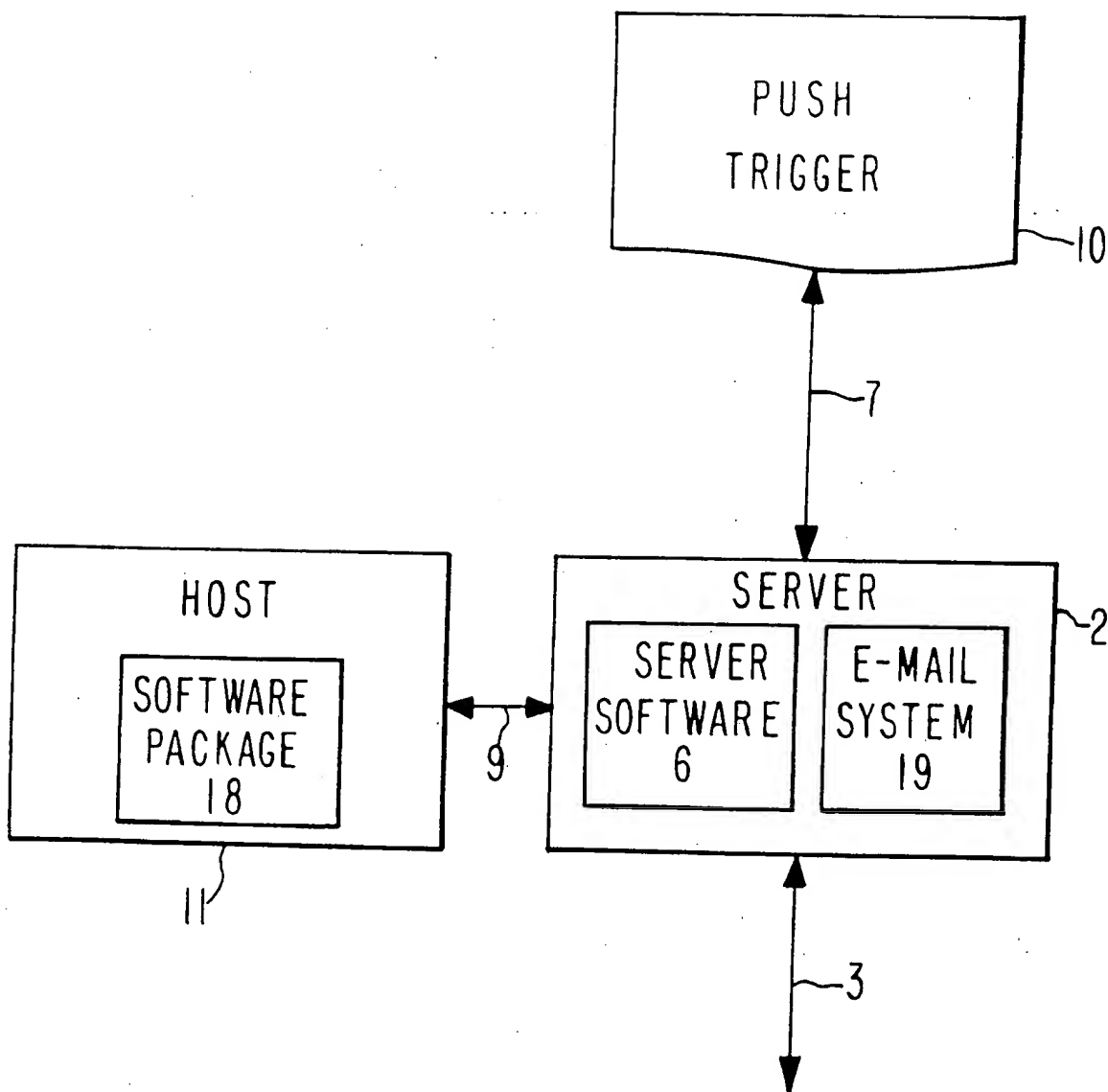
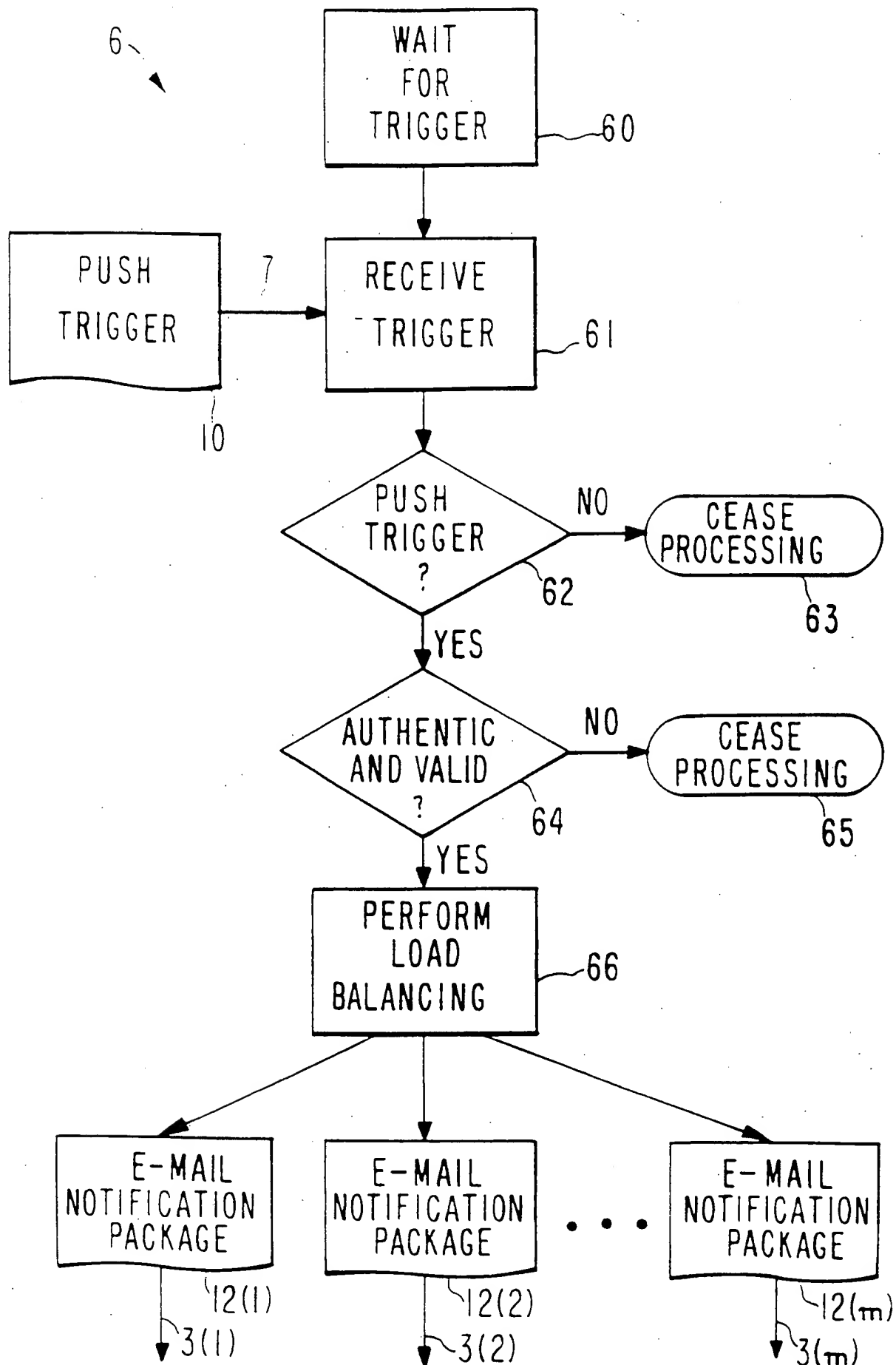


FIG.2

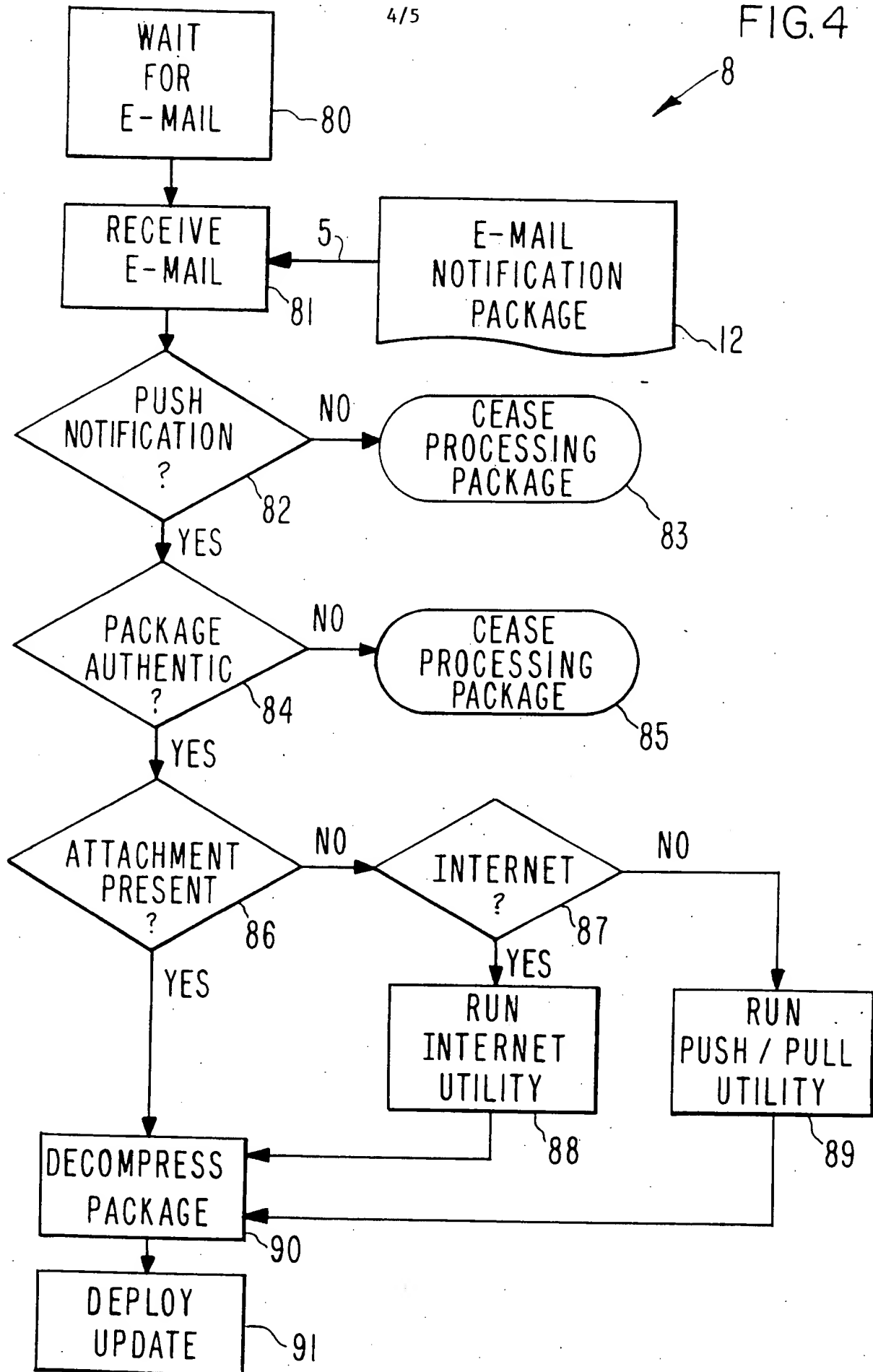
FIG. 3

3/5



4/5

FIG. 4



5/5

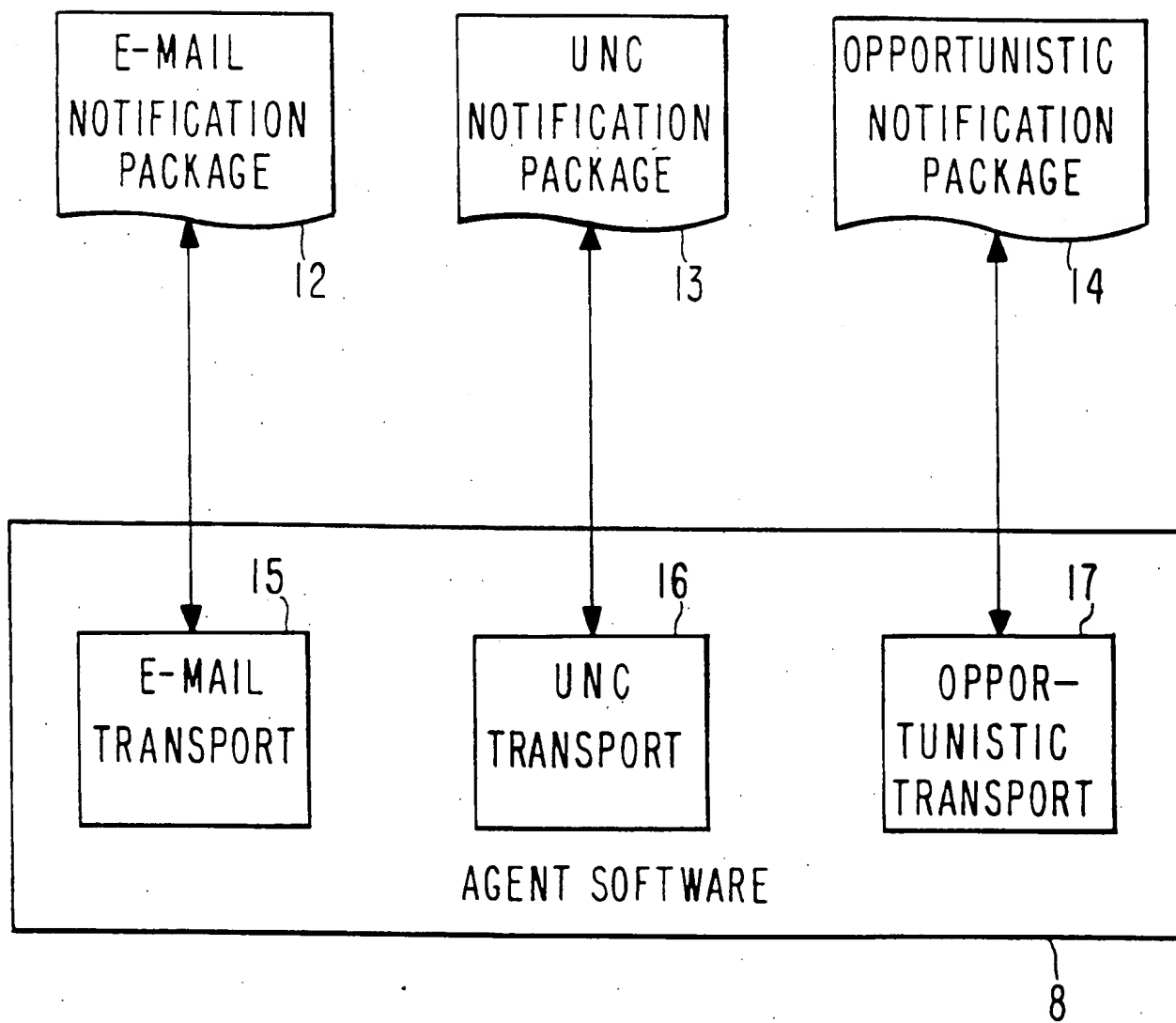


FIG. 5

PCT/US 98/08784

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F9/445

According to International Patent Classification(IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	SALAMONE S: "ELECTRONIC SOFTWARE DISTRIBUTION: DIAMOND IN THE ROUGH ESD PRODUCTS ARE A FEW YEARS AWAY FROM JOINING THE ENTERPRISE INFRASTRUCTURE, BUT THEY ALREADY SHOW PROMISE FOR YIELDING BIG SOFTWARE MANAGEMENT SAVINGS" DATA COMMUNICATIONS, vol. 22, no. 4, 1 March 1993, pages 109-110, 112, 114, 116, XP000345797 see page 109, left-hand column, line 24 - line 35 see page 116, left-hand column, line 14 - right-hand column, line 53 ---	1-17
A	US 5 421 009 A (PLATT STEPHEN M) 30 May 1995 see column 1, line 13 - column 2, line 42 see column 3, line 13 - line 33 --- -/--	1-17



Further documents are listed in the continuation of box C.



Patent family members are listed in annex.

*** Special categories of cited documents:**

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

4 August 1998

Date of mailing of the international search report

11/08/1998

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Brandt, J

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>TROKE L: "CONFIGURATION INSTALLATION AND DISTRIBUTION - A STRUCTURED APPROACH TO SOFTWARE DISTRIBUTION FOR CLIENT/SERVER (SHARE EUROPE AM93:2.2E)PASSAGE"</p> <p>CONFIGURATION INSTALLATION AND DISTRIBUTION - A STRUCTURED APPROACH TO SOFTWARE DISTRIBUTION FOR CLIENT/SERVER, 25 October 1993, pages 893-905, XP000619799</p> <p>see page 896, line 1 - page 898, line 13</p> <p>-----</p>	1-17

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5421009 A	30-05-1995	NONE	